

```

1  #include < ESP8266WiFi.h >
2  #include < WiFiClient.h >
3  #include < ESP8266WebServer.h >
4  #include < ESP8266mDNS.h >
5  #include < DNSServer.h >
6  #include < Adafruit_NeoPixel.h >
7  #define PIN D2
8
9  const char WiFiAPPSK[] = "124689";
10 const byte DNS_PORT = 53;
11 IPAddress apIP(192, 168, 1, 1);
12 DNSServer dnsServer;
13 ESP8266WebServer server(80);
14 Adafruit_NeoPixel pixels = Adafruit_NeoPixel(1, PIN, NEO_GRB + NEO_KHZ800);
15
16 void handleRoot() {
17     String red = server.arg(0); // read RGB arguments
18     String green = server.arg(1);
19     String blue = server.arg(2);
20
21     Serial.println(red.toInt()); // for TESTING
22     Serial.println(green.toInt()); // for TESTING
23     Serial.println(blue.toInt()); // for TESTING
24     String webpage;
25     webpage += "<!DOCTYPE HTML>\r\n";
26     webpage += "<html>\r\n";
27     webpage += "<head>";
28     webpage += "<meta name='mobile-web-app-capable' content='yes' />";
29     webpage += "<meta name='viewport' content='width=device-width' />";
30     webpage += "</head>";
31
32     webpage += "<body style='margin: 0px; padding: 0px;'>";
33     webpage += "<canvas id='colorspace'></canvas></body>";
34
35     webpage += "<script type='text/javascript'>";
36     webpage += "(function () {";
37     webpage += " var canvas = document.getElementById('colorspace');";
38     webpage += " var ctx = canvas.getContext('2d');";
39     webpage += " function drawCanvas() {";
40     webpage += " var colours = ctx.createLinearGradient(0, 0, window.innerWidth,
41     0);";
42     webpage += " for(var i=0; i <= 360; i+=10) {";
43     webpage += " colours.addColorStop(i/360, 'hsl(' + i + ', 100%, 50%)');";
44     webpage += " }";
45
46     webpage += " ctx.fillStyle = colours;";
47     webpage += " ctx.fillRect(0, 0, window.innerWidth, window.innerHeight);";
48     webpage += " var luminance = ctx.createLinearGradient(0, 0, 0,
49     ctx.canvas.height);";
50     webpage += " luminance.addColorStop(0, '#ffffff');";
51     webpage += " luminance.addColorStop(0.05, '#ffffff');";
52     webpage += " luminance.addColorStop(0.5, 'rgba(0,0,0,0)');";
53     webpage += " luminance.addColorStop(0.95, '#000000');";
54     webpage += " luminance.addColorStop(1, '#000000');";
55     webpage += " ctx.fillStyle = luminance;";
56     webpage += " ctx.fillRect(0, 0, ctx.canvas.width, ctx.canvas.height);";
57     webpage += " }";
58     webpage += " var eventLocked = false;";
59
60     webpage += " function handleEvent(clientX, clientY) {";
61     webpage += " if(eventLocked) {";
62     webpage += " return;";
63     webpage += " }";
64
65     webpage += " function colourCorrect(v) {";
66     webpage += " return Math.round(1023-(v*v)/64);";
67     webpage += " }";
68     webpage += " var data = ctx.getImageData(clientX, clientY, 1, 1).data;";
69     webpage += " var params = [";

```

```

68     webpage += " 'r=' + colourCorrect(data[0]),";
69     webpage += " 'g=' + colourCorrect(data[1]),";
70     webpage += " 'b=' + colourCorrect(data[2]),";
71     webpage += " ].join('&');";
72     webpage += " var req = new XMLHttpRequest();";
73     webpage += " req.open('POST', '?' + params, true);";
74     webpage += " req.send();";
75     webpage += " eventLocked = true;";
76     webpage += " req.onreadystatechange = function() {";
77     webpage += " if(req.readyState == 4) {";
78     webpage += " eventLocked = false;";
79     webpage += " }";
80     webpage += " }";
81     webpage += " }";
82     webpage += " canvas.addEventListener('click', function(event) {";
83     webpage += " handleEvent(event.clientX, event.clientY, true);";
84     webpage += " }, false);";
85     webpage += " canvas.addEventListener('touchmove', function(event) {";
86     webpage += " handleEvent(event.touches[0].clientX, event.touches[0].clientY);";
87     webpage += " }, false);";
88     webpage += " function resizeCanvas() {";
89     webpage += " canvas.width = window.innerWidth;";
90     webpage += " canvas.height = window.innerHeight;";
91     webpage += " drawCanvas();";
92     webpage += " }";
93
94     webpage += " window.addEventListener('resize', resizeCanvas, false);";
95     webpage += " resizeCanvas();";
96     webpage += " drawCanvas();";
97     webpage += " document.ontouchmove = function(e) {e.preventDefault()};";
98     webpage += " }) ();";
99     webpage += "</script><html>\r\n";
100
101     server.send(200, "text/html", webpage);
102 }
103
104 void fadeRR() {
105     for (int u = 0; u < 1024; u++) {
106         pixels.setPixelColor(0, (255 - u - 255) * -1, 0, 0);
107         pixels.show();
108         delay(1);
109     }
110     for (int u = 0; u < 1024; u++) {
111         pixels.setPixelColor(0, (u - 255) * -1, 0, 0);
112         delay(1);
113     }
114 }
115 void fadeGG() {
116     for (int u = 0; u < 1024; u++) {
117         pixels.setPixelColor(0, 0, (255 - u - 255) * -1, 0);
118         pixels.show();
119         delay(1);
120     }
121     for (int u = 0; u < 1024; u++) {
122         pixels.setPixelColor(0, 0, (u - 255) * -1, 0);
123         delay(1);
124     }
125 }
126 void fadeBB() {
127     for (int u = 0; u < 1024; u++) {
128         pixels.setPixelColor(0, 0, 0, (255 - u - 255) * -1);
129         pixels.show();
130         delay(1);
131     }
132     for (int u = 0; u < 1024; u++) {
133         pixels.setPixelColor(0, 0, 0, (u - 255) * -1);
134         delay(1);
135     }
136 }

```

```

137 void handleNotFound() {
138     //digitalWrite(led, 1);
139     String message = "File Not Found\n\n";
140     message += "URI: ";
141     message += server.uri();
142     message += "\nMethod: ";
143     message += (server.method() == HTTP_GET) ? "GET" : "POST";
144     message += "\nArguments: ";
145     message += server.args();
146     message += "\n";
147     for (uint8_t i = 0; i < server.args(); i++) {
148         message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
149     }
150     server.send(404, "text/plain", message);
151     //digitalWrite(led, 0);
152 }
153 void setup(void) {
154     pixels.begin();
155     fadeRR();
156     pixels.setPixelColor(0, 0, 0, 0);
157     fadeGG();
158     pixels.setPixelColor(0, 0, 0, 0);
159     fadeBB();
160     Serial.begin(115200);
161     setupWiFi();
162     Serial.println("");
163     delay(2000);
164     Serial.println("");
165     Serial.print("Connected to ");
166     Serial.print("IP address: ");
167     Serial.println(WiFi.localIP());
168     if (MDNS.begin("esp8266")) {
169         Serial.println("MDNS responder started");
170     }
171     server.on("/", handleRoot);
172     server.onNotFound(handleNotFound);
173     server.begin();
174     Serial.println("HTTP server started");
175 }
176 void setupWiFi() {
177     uint8_t mac[WL_MAC_ADDR_LENGTH];
178     WiFi.softAPmacAddress(mac);
179     String macID = String(mac[WL_MAC_ADDR_LENGTH - 2], HEX) +
180         String(mac[WL_MAC_ADDR_LENGTH - 1], HEX);
181     macID.toUpperCase();
182     String AP_NameString = "RahaCo Gift" + macID;
183
184     char AP_NameChar[AP_NameString.length() + 1];
185     memset(AP_NameChar, 0, AP_NameString.length() + 1);
186
187     for (int i = 0; i < AP_NameString.length(); i++)
188         AP_NameChar[i] = AP_NameString.charAt(i);
189     WiFi.mode(WIFI_AP);
190     WiFi.softAPConfig(apIP, apIP, IPAddress(255, 255, 255, 0));
191     WiFi.softAP(AP_NameChar, WiFiAPPSK);
192     dnsServer.setTTL(300);
193     dnsServer.setErrorReplyCode(DNSReplyCode::ServerFailure);
194     dnsServer.start(DNS_PORT, "gift.rahaco.net", apIP);
195 }
196 void loop(void) {
197     dnsServer.processNextRequest();
198     server.handleClient();
199 }

```